M5 Forecasting Competition Write-up

Introduction

The M5 Forecasting Competition challenges participants to forecast daily sales for Walmart's inventory across multiple departments, stores, and product levels. Our team explored a progression of modeling approaches, beginning with LightGBM for baseline results, advancing through recurrent neural networks (LSTM and GRU), and finally implementing a Seq2Seq GRU model for efficient sequence forecasting. The goal was to evaluate performance, scalability, and modeling flexibility under limited computational resources.

Table 1: Explored Models and Hyper-Parameters

Model	Framework	Purpose	Key Hyperparameters
LightGBM	LightGBM	Baseline with engineered features	Tweedie loss (variance = 1.1), learning_rate=0.05, store-level splits
LSTM	Keras	Time series with memory	time_steps=14, units=128, dropout=0.2, batch_size=32
GRU	Keras	Lightweight time series model	time_steps=14, units=64, dropout=0.2, batch_size=32
Seq2Seq (GRU)	Keras	One-shot 28-day forecasting	encoder/decoder units=64, time_steps=14, forecast_steps=28

Hyperparameter tuning was done manually based on observed training loss and GPU limitations. Batch sizes and model widths were optimized to prevent out-of-memory errors.

Table 2: Feature Engineering Summary

Feature Type	Description	Used In	Rationale
Lag features	sales_lag_7, sales_lag_28	LightGBM	Capture repeated weekly demand
Rolling statistics	rolling_mean_7, rolling_std_28	LightGBM	Reflect demand volatility and trends
Price signals	price_norm, price_momentum	LightGBM	Influence of pricing on demand
Time-based	weekday, weekend, month, year	LightGBM	Seasonality and shopping patterns

Feature Type	Description	Used In	Rationale
Event flags	Binary next-day event indicator	RNN/LSTM/GRU	Account for calendar-based demand spikes
Sliding sales windows	Last 14 days of raw sales per item	RNN/LSTM/GRU	Allow network to capture short-term sales trends
Transposed sales matrix	Days as rows, items as columns	Seq2Seq GRU	Simplifies input structure, improves memory efficiency

For LSTM/GRU models, the focus was on minimizing preprocessing to avoid memory overhead. We engineered time-based input tensors from the raw sales matrix and selectively included binary event indicators to help detect pre-holiday surges or weekly demand shifts.

Why we did not use all available features: Price-based features and rolling statistics, while powerful in tree-based models, were not included in LSTM/GRU/Seq2Seq models due to:

- Increased memory footprint during sequence modeling
- Difficulty aligning auxiliary features across multiple time steps
- Focus on simplicity and scalability, especially given GPU limitations

The Seq2Seq GRU used a transposed format with no additional features to maintain a minimalistic design suitable for one-shot multi-day forecasts under memory constraints.

Table 3: Model Performance Comparison

Submission File	Model	Private Score	Notes
submission.csv	LightGBM	5.39063	High error, baseline model
submission_LSTM_*.csv	LSTM	0.98608	Better than LightGBM, but overfit on train data
submission_GRU_*.csv	GRU	**0.78671**	**Best performance**, balanced fit
simple_seq2seq_*.csv	Seq2Seq GRU	0.99343	Efficient but underfit due to lack of metadata

GRU provided the best generalization and lowest error, closely followed by LSTM. Seq2Seq was fastest to run but slightly underperformed.

Screenshot: Kaggle Submission Results

Below is a screenshot of our best-performing GRU model from the Kaggle submission system:

Conclusion

Our exploration highlights that while feature-based models like LightGBM provide interpretable and scalable solutions, neural models—especially GRUs—offer a better balance of performance and efficiency in sequential forecasting tasks. The Seq2Seq architecture shows great promise for fast deployment, and with additional feature integration, could rival the best-performing GRU models in future iterations.

This competition served as an in-depth exercise in time series modeling, feature selection, architecture tuning, and managing performance under resource constraints.